Parallel and Concurrent Security of the HB and HB$^+$ Protocols

Jonathan Katz[⋆] and Ji Sun Shin[⋆⋆]

Dept. of Computer Science
University of Maryland
{jkatz,sunny}@cs.umd.edu

Abstract. Juels and Weis (building on prior work of Hopper and Blum) propose and analyze two shared-key authentication protocols — HB and HB$^+$ — whose extremely low computational cost makes them attractive for low-cost devices such as radio-frequency identification (RFID) tags. Security of these protocols is based on the conjectured hardness of the "learning parity with noise" (LPN) problem: the HB protocol is proven secure against a passive (eavesdropping) adversary, while the HB$^+$ protocol is proven secure against active attacks.

Juels and Weis prove security of these protocols only for the case of *sequential* executions, and explicitly leave open the question of whether security holds also in the case of *parallel* or *concurrent* executions. In addition to guaranteeing security against a stronger class of adversaries, a positive answer to this question would allow the HB$^+$ protocol to be parallelized, thereby substantially reducing its round complexity.

Adapting a recent result by Regev, we answer the aforementioned question in the affirmative and prove security of the HB and HB$^+$ protocols under parallel/concurrent executions. We also give what we believe to be substantially *simpler* security proofs for these protocols which are more *complete* in that they explicitly address the dependence of the soundness error on the number of iterations.

1 Introduction

Low-cost, severely resource-constrained devices such as radio-frequency identification (RFID) tags or sensor nodes demand extremely efficient algorithms and protocols. Securing such devices is a challenge since, in many cases, "traditional" cryptographic protocols are simply too computationally-intensive to be utilized. With this motivation in mind, Juels and Weis [20] — building upon work of Hopper and Blum [18, 19] — investigate two highly-efficient, shared-key (unidirectional) authentication protocols suitable for an RFID *tag* identifying itself to a tag *reader*. (We will sometimes refer to the tag as a *prover* and the tag reader

⋆ This research was supported by NSF Trusted Computing grants #0310499 and #0310751, and NSF CAREER award #0447075.
⋆⋆ Supported by NSF Trusted Computing grant #0310499.

as a *verifier*.) These protocols are extremely lightweight, requiring both parties to perform only a relatively small number of primitive bit-wise operations such as "XOR" and "AND," and can thus be implemented using fewer than the 5-10K gates required to implement even a block cipher such as DES or AES [20].

The two protocols studied by Juels and Weis are both proven secure via reduction to the "learning parity with noise" (LPN) problem [4–6, 9, 17, 21, 18, 19, 25]; a formal definition of this problem as well as evidence for its difficulty are reviewed in Section 2.1. The first protocol (called the HB protocol [18, 19]) is proven secure against a *passive* (eavesdropping) adversary, while the second (called HB^+) is proven secure against the stronger class of *active* adversaries. In each case, Juels and Weis focus on a single, "basic authentication step" of the protocol and prove that a computationally-bounded adversary cannot succeed in impersonating a tag in this case with probability noticeably better than 1/2; that is, a single iteration of the protocol has *soundness error* 1/2. The implicit assumption (though see below) is that repeating these "basic authentication steps" sufficiently-many times yields a protocol with negligible soundness error.

Difficulties and limitations. There are, however, some subtle limitations of the security proofs given by Juels and Weis. Most serious, perhaps, is a difficulty explicitly highlighted by Juels and Weis and regarded by them as a potential barrier to usage of the HB^+ protocol in practice [20, Section 6]: the proof of security for HB^+ requires that the adversary's interactions with the tag (i.e., when the adversary is impersonating a tag reader) be *sequential*. Besides leaving in question the security of HB^+ under *concurrent* executions, this also means that the HB^+ protocol itself (which, recall, consists of sufficiently-many repetitions of an underlying basic authentication step) requires very high round complexity since the multiple iterations of the basic authentication step cannot be *parallelized* but must instead be performed sequentially. The difficulty and importance of proving security of various identification protocols under concurrent or parallel composition is well-understood, and many results are known: for example, the (black-box) zero-knowledge property of an identification protocol is not preserved under parallel [14] or concurrent [8] composition (though it is preserved under sequential composition [16]), whereas witness indistinguishability *is* preserved in these cases [11]. Unfortunately, the HB^+ protocol is not known to satisfy either zero knowledge or witness indistinguishability and so such results are of no help here.

An additional difficulty, not explicitly mentioned in [20], is that it is unclear what is the exact relationship between the soundness error and the number of repetitions of the basic authentication step; this is true for both the HB and HB^+ protocols, regardless of whether the repetitions are carried out in parallel or sequentially.[1] This is related to the more general question of "when is solving multiple instances of a problem more difficult than solving a single instance?"

[1] Indeed, Juels and Weis only prove soundness 1/2 for a basic authentication step and never make any claims regarding the security of multiple iterations (for either HB or HB^+); this indicates that those authors also recognized the difficulty of characterizing the dependence of soundness on the number of iterations.

(i.e., *hardness amplification*) which has been studied in many contexts [26, 15, 3, 13, 24, 7] and turns out to be surprisingly non-trivial to answer. Unfortunately, there does not seem to be any prior work that applies in our setting. Specifically:

- For the HB and HB^+ protocols it is not possible to efficiently verify whether a given transcript is "successful" without possession of the secret key; thus, Yao's "XOR-lemma" [26, 15] and related techniques that require efficient verifiability do not apply.
- Work on hardness amplification for "weakly-verifiable puzzles" [7] does not apply either. Although the HB/HB^+ protocols can be viewed as efficiently-verifiable puzzles, hardness amplification in [7] is only proved for *completely independent* instances of the "puzzle." In particular, then, the work of [7] implies that running the basic authentication step of the HB protocol n times *using n independent keys* yields soundness (roughly) $1/2^n$, but says nothing about running n iterations using the *same* key (which is the case we are interested in).
- The HB/HB^+ protocols are *computationally*-sound only, and thus known results [13, Appendix C] [24] on soundness reduction for interactive proof systems (which apply only when soundness holds even against an all-powerful cheating prover) do not apply either.
- Bellare, et al. [3] study soundness reduction in computationally-sound protocols, and show a positive result [3, Sect. 4] for the case of protocols running in 3 rounds. Unfortunately, their result is specifically stated to apply *only* when the verifier does not hold a secret key (or, more generally, only when the verifier does not share state across different iterations). As in the case of weakly-verifiable puzzles, then, this result is of no help when the same secret key is used across all iterations.

An additional difficulty in our setting is that the verifier is supposed to accept even when some iterations have not been answered successfully; indeed, crucial to both the HB and HB^+ protocols is that the honest prover injects "noise" into its answers and so even the honest prover does not succeed with probability 1. This was not explicitly addressed in the security proofs of [20], either, and introduces additional complications.

1.1 Our Contributions

In this work we address the difficulties and open questions mentioned above, and show the following results: (1) the HB^+ protocol remains secure under arbitrary concurrent interactions of the adversary with the honest prover/tag, and so in particular the iterations of the HB^+ protocol can be parallelized; furthermore, (2) our security proofs explicitly incorporate the dependence of the soundness error on the number of iterations as well as the error introduced by the honest prover.

Besides the results themselves, we expect that the techniques and proofs we give here will be of independent interest for future work on cryptographic applications of the LPN problem. Our main technical tool is a result due to

Regev [25] (see also [5]) showing that the hardness of the LPN problem implies the pseudorandomness of a certain distribution. Using this, we give proofs which we believe are substantially *simpler* than those given in [20], and also more *complete* (in that, in contrast to [20], they explicitly deal with the dependence of soundness on the number of iterations and also the issues arising due to non-perfect completeness).

1.2 Additional Discussion

The problem of secure authentication using a shared, secret key is by now well-understood, and many widely-known solutions based on, e.g., block ciphers are available. We stress that the aim of the line of research considered here, as in [20], is to develop protocols which are *exceptionally* efficient while still guaranteeing some useful level of (provable) security. The estimates from [20] are that 5,000–$10{,}000^{+}$ gates are needed for block-cipher implementations, whereas a typical RFID tag may only have 2,000 gates that can be dedicated to security. Moore's Law will not necessarily help here, either: as pointed out in [20], there is intense pressure to keep prices for RFID tags low; as computational power per fixed unit of currency increases, the trend has been to reduce the cost of tags and thus expand their application domain rather than to increase their computational power while keeping costs fixed. In short, there seems to be "little effective change in tag resources for some time to come, and thus a pressing need for new lightweight primitives" [20].

Gilbert, et al. [12] have recently shown a man-in-the-middle attack on the HB^{+} protocol. Although their attack would be debilitating if carried out successfully, the possibility of such an attack does not mean that it is now useless to explore the security of the $\mathrm{HB}/\mathrm{HB}^{+}$ protocols in weaker attack models! (Indeed, only recently have man-in-the-middle attacks on identification protocols been formally considered in general [2], yet certainly research in the area conducted up to that point is not valueless.) There will always be some tradeoff between efficiency and security, and our work can be viewed as mapping out where the $\mathrm{HB}/\mathrm{HB}^{+}$ protocols lie on this spectrum. Moreover, Juels and Weis [20, Appendix A] note that the man-in-the-middle attack of [12] does not apply in a *detection-based* system where numerous failed authentication attempts immediately raise an alarm. Furthermore, especially in the case of RFID (where communication is inherently short range), it appears much more difficult to mount a man-in-the-middle attack than an active attack.[2] The reader is referred to the work of Wool, et al. [22, 23], for an illuminating discussion on the feasibility of man-in-the-middle attacks in RFID systems.

---

[2] Though there have been claims of being able to read some RFID tags over as much as 69 feet [1], the maximum distance from which many commonly-used cards can be read appears to be almost two orders of magnitude lower [22]. Note further that a man-in-the-middle attack requires the ability to *send* data to the tag (and reader).

2 Definitions and Preliminaries

We formally define the LPN problem and state and prove the main technical lemma on which we rely. We also define our notion(s) of security for identification; these are standard, but some complications arise due to the fact that the HB/HB$^+$ protocols do not have perfect completeness.

2.1 The LPN Problem

View k as a security parameter. If $\mathbf{s}, \mathbf{a}_1, \ldots, \mathbf{a}_\ell$ are binary vectors of length k, let $z_i = \langle \mathbf{s}, \mathbf{a}_i \rangle$ denote the dot product of $\mathbf{s}$ and $\mathbf{a}_i$ (modulo 2). Given the values $\mathbf{a}_1, z_1, \ldots, \mathbf{a}_\ell, z_\ell$ for randomly-chosen $\{\mathbf{a}_i\}$ and $\ell = O(k)$, it is possible to efficiently solve for $\mathbf{s}$ using standard linear-algebraic techniques. However, in the presence of *noise* where each z_i is flipped (independently) with probability ε, finding $\mathbf{s}$ becomes much more difficult. We refer to the problem of learning $\mathbf{s}$ in this latter case as *the LPN problem*.

For the formal definition, let Ber_ε be the Bernoulli distribution with parameter $\varepsilon \in (0, \frac{1}{2})$ (so if $\nu \sim \mathsf{Ber}_\varepsilon$ then $\Pr[\nu = 1] = \varepsilon$ and $\Pr[\nu = 0] = 1 - \varepsilon$), and let $A_{\mathbf{s},\varepsilon}$ be the distribution defined by:

$$\left\{\mathbf{a} \leftarrow \{0,1\}^k; \nu \leftarrow \mathsf{Ber}_\varepsilon : (\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle \oplus \nu)\right\}.$$

Also let $A_{\mathbf{s},\varepsilon}$ denote an oracle which outputs (independent) samples according to this distribution. Algorithm M is said to (t, q, δ)*-solve the* LPN_ε *problem* if

$$\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : M^{A_{\mathbf{s},\varepsilon}}(1^k) = \mathbf{s}\right] \geq \delta,$$

and furthermore M runs in time at most t and makes at most q queries to its oracle.[3] In asymptotic terms, in the standard way, the LPN_ε problem is "hard" if every probabilistic polynomial-time algorithm solves the LPN_ε problem with only negligible probability (where the algorithm's running time and success probability are functions of k).

Note that ε is usually taken to be a fixed constant independent of k, as will be the case here. The value of ε to use depends on a number of tradeoffs and design decisions: although, roughly speaking, the LPN_ε problem becomes "harder" as ε increases, a larger value of ε also implies that the honest prover is rejected more often (as will become clear when we describe the HB/HB$^+$ protocols, below). In any case, our results are meaningful for all $\varepsilon \in (0, \frac{1}{4})$. For concreteness, the reader can think of $\varepsilon \approx \frac{1}{8}$.

The hardness of the LPN_ε problem (for constant $\varepsilon \in (0, \frac{1}{2})$) has been studied in many previous works. It can be formulated also as the problem of decoding a random linear code [4, 25], and is $\mathcal{NP}$-complete [4] as well as hard to approximate within a factor better than 2 (where the optimization problem is phrased

[3] Our formulation of the LPN problem follows, e.g., [25]; the formulation in, e.g., [20] allows M to output any $\mathbf{s}$ satisfying $\geq (1 - \varepsilon)$ fraction of the equations returned by $A_{\mathbf{s},\varepsilon}$. It is easy to see that for q large enough these formulations are essentially equivalent as with overwhelming probability there will be a unique such $\mathbf{s}$.

as finding an $\mathbf{s}$ satisfying the most equations) [17]. These worst-case hardness results are complemented by numerous studies of the average-case hardness of the problem [5, 6, 9, 21, 18, 19, 25]. Most relevant for our purposes is that the current best-known algorithm for solving the LPN_ε problem [6] requires $t, q = 2^{\Theta(k/\log k)}$. We refer the reader to [20, Appendix D] for more exact estimates of the running time of this algorithm, as well as suggested practical values for k.

2.2 A Technical Lemma

In this section we prove a key technical lemma: hardness of the LPN_ε problem implies "pseudorandomness" of $A_{\mathbf{s},\varepsilon}$. Specifically, let U_{k+1} denote the uniform distribution on $(k+1)$-bit strings. The following lemma shows that oracle access to $A_{\mathbf{s},\varepsilon}$ (for randomly-chosen $\mathbf{s}$) is indistinguishable from oracle access to U_{k+1}. A proof of the following is essentially in [25, Sect. 4], although we have fleshed out some of the details and worked out the concrete parameters of the reduction.

Lemma 1. *Say there exists an algorithm D making q oracle queries, running in time t, and such that*

$$\left|\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : D^{A_{\mathbf{s},\varepsilon}}(1^k) = 1\right] - \Pr\left[D^{U_{k+1}}(1^k) = 1\right]\right| \geq \delta.$$

Then there exists an algorithm M making $q' = O\left(q \cdot \delta^{-2} \log k\right)$ oracle queries, running in time $t' = O\left(t \cdot k\delta^{-2} \log k\right)$, and such that

$$\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : M^{A_{\mathbf{s},\varepsilon}}(1^k) = \mathbf{s}\right] \geq \delta/4.$$

(Various tradeoffs are possible between the number of queries/running time of M and its success probability in solving LPN_ε; see [25, Sect. 4]. We aimed for simplicity in the proof rather than trying to optimize parameters.)

Proof. Set $N = O\left(\delta^{-2} \log k\right)$. Algorithm $M^{A_{\mathbf{s},\varepsilon}}(1^k)$ proceeds as follows:

1. M chooses random coins ω for D and uses these for the remainder of its execution.
2. M runs $D^{U_{k+1}}(1^k; \omega)$ for a total of N times to obtain an estimate p for the probability that D outputs 1 in this case. (The probability here is over the responses from the oracle.)
3. M obtains $q \cdot N$ samples $\{(\mathbf{a}_{1,j}, z_{1,j})\}_{j=1}^q, \ldots, \{(\mathbf{a}_{N,j}, z_{N,j})\}_{j=1}^q$ from $A_{\mathbf{s},\varepsilon}$. Then for $i \in [k]$:
   (a) Run $D(1^k; \omega)$ for a total of N times, each time using a fresh set of samples $\{(\mathbf{a}_j, z_j)\}_{j=1}^q$ to answer the q oracle queries of D. Answer the j^{th} oracle query of D in each iteration by choosing a random bit c_j and returning $(\mathbf{a}_j \oplus (c_j \cdot \mathbf{e}_i),\ z_j)$, where $\mathbf{e}_i$ is the vector with 1 at position i and 0s elsewhere. Obtain an estimate p_i for the probability that D outputs 1 in this case.
   (b) If $|p_i - p| \geq \delta/4$ set $s'_i = 0$; else set $s'_i = 1$.
4. Output $\mathbf{s}' = (s'_1, \ldots, s'_k)$.

Let us analyze the behavior of M. First note that, by standard averaging argument, with probability at least $\delta/2$ over choice of $\mathbf{s}$ and random coins ω it holds that

$$\left|\Pr\left[D^{A_{\mathbf{s},\varepsilon}}(1^k;\omega)=1\right]-\Pr\left[D^{U_{k+1}}(1^k;\omega)=1\right]\right|\geq\delta/2, \tag{1}$$

where the probabilities are taken over the answers D receives from its oracle. We restrict our attention to $\mathbf{s},\omega$ for which Eq. (1) holds and show that in this case M outputs $\mathbf{s}'=\mathbf{s}$ with probability at least $1/2$. The theorem follows.

By our choice of N we have that

$$\left|\Pr\left[D^{U_{k+1}}(1^k;\omega)=1\right]-p\,\right|\leq\delta/16 \tag{2}$$

except with probability at most $O(1/k)$. Next focus on a particular iteration i of steps 3(a) and 3(b). Letting hyb_i denote the distribution of the answers returned to D in this iteration, we again have

$$\left|\Pr\left[D^{\mathsf{hyb}_i}(1^k;\omega)=1\right]-p_i\,\right|\leq\delta/16 \tag{3}$$

except with probability at most $O(1/k)$. Applying a union bound (and setting parameters appropriately) we see that with probability at least $1/2$ Eqs. (2) and (3) hold (the latter for all $i\in[k]$), and so we assume this to be the case for the rest of the proof.

We claim that if $s_i=0$ then $\mathsf{hyb}_i=A_{\mathbf{s},\varepsilon}$, while if $s_i=1$ then $\mathsf{hyb}_i=U_{k+1}$. To see this note that when $s_i=0$ the answer $(\mathbf{a}_j\oplus(c_j\cdot\mathbf{e}_i),\,z_j)$ returned to D is distributed exactly according to $A_{\mathbf{s},\varepsilon}$ since $\langle\mathbf{s},\,\mathbf{a}_j\oplus(c_j\cdot\mathbf{e}_i)\rangle=\langle\mathbf{s},\mathbf{a}_j\rangle=z_j$. On the other hand, if $s_i=1$ then $z_j=\langle\mathbf{s},\mathbf{a}_j\rangle$ is independent of $\mathbf{a}_j\oplus(c_j\cdot\mathbf{e}_i)$ since c_j is random (and unknown to D).

It follows that if $s_i=0$ then

$$\left|\Pr\left[D^{\mathsf{hyb}_i}(1^k;\omega)=1\right]-\Pr\left[D^{U_{k+1}}(1^k;\omega)=1\right]\right|\geq\delta/2$$

(by Eq. (1)), and so $|p_i-p\,|\geq\frac{\delta}{2}-2\cdot\frac{\delta}{16}=\frac{3\delta}{8}$ (by Eqs. (2) and (3)) and $s_i'=0=s_i$. When $s_i=1$ then

$$\Pr\left[D^{\mathsf{hyb}_i}(1^k;\omega)=1\right]=\Pr\left[D^{U_{k+1}}(1^k;\omega)=1\right],$$

and so $|p_i-p\,|\leq2\cdot\frac{\delta}{16}=\frac{\delta}{8}$ (again using Eqs. (2) and (3)) and $s_i'=1=s_i$. Since this holds for all $i\in[k]$, we conclude that $\mathbf{s}'=\mathbf{s}$.

2.3 Overview of the HB/HB$^+$ Protocols, and Security Definitions

The HB and HB$^+$ protocols as analyzed here consist of n *parallel* iterations of a "basic authentication step." We describe the basic authentication step for the HB protocol, and defer a discussion of the HB$^+$ protocol to Section 3.2. In the HB protocol, a tag $\mathcal{T}$ and a reader $\mathcal{R}$ share a random secret key $\mathbf{s}\in\{0,1\}^k$; a basic authentication step consists of the reader sending a random challenge $\mathbf{a}\in\{0,1\}^k$ to the tag, which replies with $z=\langle\mathbf{s},\mathbf{a}\rangle\oplus\nu$ for $\nu\sim\mathsf{Ber}_\varepsilon$. The reader

$$
\begin{array}{ccc}
\underline{\mathcal{T}(\mathbf{s},\varepsilon)} & & \underline{\mathcal{R}(\mathbf{s})} \\
 & \xleftarrow{\quad \mathbf{a} \quad} & \mathbf{a} \leftarrow \{0,1\}^k \\
\nu \leftarrow \mathsf{Ber}_\varepsilon & & \\
z := \langle \mathbf{s}, \mathbf{a} \rangle \oplus \nu & \xrightarrow{\quad z \quad} & \\
 & & \text{verify: } z \stackrel{?}{=} \langle \mathbf{s}, \mathbf{a} \rangle
\end{array}
$$

Fig. 1. The basic authentication step of the HB protocol.

can then verify whether the response z of the tag satisfies $z \stackrel{?}{=} \langle \mathbf{s}, \mathbf{a} \rangle$; we say the iteration is *successful* if this is the case. See Figure 1.

Even for an honest tag a basic iteration is unsuccessful with probability ε. For this reason, a reader accepts upon completion of all n iterations of the basic authentication step as long as $\approx \varepsilon \cdot n$ of these iterations were unsuccessful. More precisely, let l, u be such that $\mathsf{l} \leq \varepsilon \cdot n \leq \mathsf{u}$; then the reader accepts as long as the number of unsuccessful iterations lies in the range $[\mathsf{l}, \mathsf{u}]$. Since $\varepsilon \cdot n$ is the expected number of unsuccessful iterations for an honest tag, the completeness error ε_c (i.e., the probability that an honest tag is rejected) can be calculated via a Chernoff bound.[4] Overall, then, the entire HB protocol is parameterized by $\varepsilon, \mathsf{l}, \mathsf{u}$, and n.

Observe that by sending random answers in each of the n iterations, an adversary trying to impersonate a valid tag succeeds with probability

$$
\delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n} \stackrel{\text{def}}{=} 2^{-n} \cdot \sum_{i=\mathsf{l}}^{\mathsf{u}} \binom{n}{i};
$$

that is, $\delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}$ is the *best* possible soundness error we can hope to achieve for the given setting of the parameters. Our definitions of security will be expressed in terms of the adversary's ability to do better than this. Looking at asymptotic security (taking k as a security parameter), note that for any constant $\varepsilon < 1/2$ it is easy to find functions $\mathsf{l}, \mathsf{u}, n$ of k such that $n = O(k)$ and furthermore both the completeness error ε_c and the "best achievable" soundness error $\delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}$ are negligible.

Let $\mathcal{T}^{\mathsf{HB}}_{\mathbf{s},\varepsilon,n}$ denote the tag algorithm in the HB protocol when the tag holds secret key $\mathbf{s}$ (note that the tag algorithm is independent of l, u), and let $\mathcal{R}^{\mathsf{HB}}_{\mathbf{s},\varepsilon,\mathsf{l},\mathsf{u},n}$ similarly denote the algorithm run by the tag reader. We denote a complete execution of the HB protocol between a party $\hat{\mathcal{T}}$ and the reader $\mathcal{R}$ by $\left\langle \hat{\mathcal{T}}, \mathcal{R}^{\mathsf{HB}}_{\mathbf{s},\varepsilon,\mathsf{l},\mathsf{u},n} \right\rangle$ and say this equals 1 iff the reader accepts.

[4] Note in particular that if u is set to *exactly* $\varepsilon \cdot n$ then the completeness error will be rather high. One can imagine changing the protocol so that the tag introduces *at most* $\varepsilon \cdot n$ errors; see Section 4 for discussion of this point.

For a passive attack on the HB protocol, we imagine an adversary $\mathcal{A}$ running in two stages: in the first stage the adversary obtains q transcripts[5] of (honest) executions of the protocol by interacting with an oracle $\mathsf{trans}^{\mathsf{HB}}_{\mathbf{s},\varepsilon,n}$ (this models eavesdropping); in the second stage, the adversary interacts with the reader and tries to impersonate the tag. We define the adversary's advantage as

$$\begin{aligned}&\mathsf{Adv}^{passive}_{\mathcal{A},\mathsf{HB}}(\varepsilon,\mathsf{l},\mathsf{u},n) \stackrel{\text{def}}{=}\\ &\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k; \mathcal{A}^{\mathsf{trans}^{\mathsf{HB}}_{\mathbf{s},\varepsilon,n}}(1^k) : \left\langle \mathcal{A}, \mathcal{R}^{\mathsf{HB}}_{\mathbf{s},\varepsilon,\mathsf{l},\mathsf{u},n}\right\rangle = 1\right] - \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}.\end{aligned}$$

As we will describe in Section 3.2, the HB$^+$ protocol uses two keys $\mathbf{s}_1, \mathbf{s}_2$. We let $\mathcal{T}^{\mathsf{HB}^+}_{\mathbf{s}_1,\mathbf{s}_2,\varepsilon,n}$ denote the tag algorithm in this case, and let $\mathcal{R}^{\mathsf{HB}^+}_{\mathbf{s}_1,\mathbf{s}_2,\varepsilon,\mathsf{l},\mathsf{u},n}$ denote the algorithm run by the tag reader. For the case of an active attack on the HB$^+$ protocol, we again imagine an adversary running in two stages: in the first stage the adversary interacts at most q times with the honest tag algorithm (with concurrent executions allowed), while in the second stage the adversary interacts only with the reader.[6] The adversary's advantage in this case is

$$\begin{aligned}&\mathsf{Adv}^{active}_{\mathcal{A},\mathsf{HB}^+}(\varepsilon,\mathsf{l},\mathsf{u},n) \stackrel{\text{def}}{=}\\ &\Pr\left[\mathbf{s}_1,\mathbf{s}_2 \leftarrow \{0,1\}^k; \mathcal{A}^{\mathcal{T}^{\mathsf{HB}^+}_{\mathbf{s}_1,\mathbf{s}_2,\varepsilon,n}}(1^k) : \left\langle \mathcal{A}, \mathcal{R}^{\mathsf{HB}^+}_{\mathbf{s}_1,\mathbf{s}_2,\varepsilon,\mathsf{l},\mathsf{u},n}\right\rangle = 1\right] - \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}.\end{aligned}$$

We remark that in both the HB and HB$^+$ protocols, the tag reader's actions are independent of the secret key(s) it holds except for its final decision whether or not to accept. So, allowing the adversary to interact with the reader multiple times (even concurrently) does not give the adversary much additional advantage (other than the fact that, as usual, the probability that the adversary succeeds in at least one impersonation attempt scales linearly with the number of attempts).

3 Proofs of Security for the HB and HB$^+$ Protocols

3.1 Security of the HB Protocol against Passive Attacks

Recall from the previous section that we parameterize the HB protocol by ε (a measure of the noise introduced by the tag), l, u (which determine the completeness error ε_c as well as the best achievable soundness δ^*), and n (the number of iterations of the basic authentication step given in Figure 1). We stress that these n iterations are run *in parallel*, and so the entire protocol requires only two rounds.

[5] Following [18–20], a transcript comprises only the messages exchanged between the parties and does not include the reader's decision of whether or not to accept. If the adversary is given this additional information, the adversary's advantage may increase by (at most) an additive factor of $q \cdot \varepsilon_c$.

[6] As we have already noted, this is the "classical" notion of security against active attacks which does not take into account man-in-the-middle attacks.

The following result characterizes security of the HB protocol against passive attack. This can be compared to [20, Lemma 1], where Juels and Weis prove security for a single iteration of the HB protocol (i.e., they fix $n = 1$) and do not explicitly take the non-zero completeness error into account (this is taken into account in the following via the dependence on l, u).

Theorem 1. *Say there exists an adversary $\mathcal{A}$ eavesdropping on q executions of the HB protocol, running in time t, and achieving $\mathsf{Adv}^{passive}_{\mathcal{A},\mathsf{HB}}(\varepsilon, \mathsf{l}, \mathsf{u}, n) \geq \delta$. Then there exists an algorithm D making $(q+1) \cdot n$ oracle queries, running in time $O(t)$, and such that*

$$\left|\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : D^{A_{\mathbf{s},\varepsilon}}(1^k) = 1\right] - \Pr\left[D^{U_{k+1}}(1^k) = 1\right]\right| \\ \geq \delta + \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n} - \varepsilon_c - 2^{-n} \cdot \sum_{i=0}^{2\,\mathsf{u}} \binom{n}{i}.$$

Asymptotically, for any $\varepsilon < \frac{1}{4}$ and $n = \Theta(k)$ all terms of the above expression (other than δ) are negligible for appropriate choice of l, u. We thus conclude that the HB protocol is secure (for $n = \Theta(k)$ and appropriate choice of l, u) assuming the hardness of the LPN_ε problem.

Proof. Algorithm D, given access to an oracle returning $(k+1)$-bit strings $(\mathbf{a}, z)$, proceeds as follows:

1. D runs the first phase of $\mathcal{A}$. Each time $\mathcal{A}$ requests to view a transcript of the protocol, D obtains n samples $\{(\mathbf{a}_i, z_i)\}_{i=1}^n$ from its oracle and returns these to $\mathcal{A}$.
2. When $\mathcal{A}$ is ready for the second phase, D again obtains n samples $\{(\bar{\mathbf{a}}_i, \bar{z}_i)\}_{i=1}^n$ from its oracle. D then sends the challenge $(\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_n)$ to $\mathcal{A}$ and receives in return a response $Z' = (z'_1, \ldots, z'_n)$.
3. D outputs 1 iff $\bar{Z} = (\bar{z}_1, \ldots, \bar{z}_n)$ and Z' differ in at most $2\mathsf{u}$ entries.

When D's oracle is U_{k+1}, it is clear that D outputs 1 with probability exactly $2^{-n} \cdot \sum_{i=0}^{2\mathsf{u}} \binom{n}{i}$ since $\bar{Z}$ is in this case uniformly distributed and independent of everything else. On the other hand, when D's oracle is $A_{\mathbf{s},\varepsilon}$ then the transcripts D provides to $\mathcal{A}$ during the first phase of $\mathcal{A}$'s execution are distributed identically to real transcripts in an execution of the HB protocol. Let $Z^* \stackrel{\text{def}}{=} (\langle \mathbf{s}, \bar{\mathbf{a}}_1 \rangle, \ldots, \langle \mathbf{s}, \bar{\mathbf{a}}_n \rangle)$ be the vector of "correct" answers to the challenge $(\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_n)$ sent by D in the second phase. Then with probability at least $\delta + \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}$ it holds that Z' and Z^* differ in at most u entries (since $\mathcal{A}$ successfully impersonates the tag with this probability). Also, since $\bar{Z}$ is distributed exactly as the answers of an honest tag, $\bar{Z}$ and Z^* differ in at most u positions except with probability at most ε_c. It follows that with probability at least $\delta + \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n} - \varepsilon_c$ the vectors Z' and $\bar{Z}$ differ in at most $2\mathsf{u}$ entries, and so D outputs 1 with at least this probability. □

The above result provides a useful security guarantee only when $\varepsilon < 1/4$, since when $\varepsilon \geq 1/4$ then $2\mathsf{u} \geq 2\varepsilon \cdot n \geq n/2$ and so $2^{-n} \cdot \sum_{i=0}^{2\,\mathsf{u}} \binom{n}{i} \geq 1/2$. We also note that the concrete security reduction obtained above leaves much to be desired, and in particular it is not clear whether useful levels of security are achieved for reasonably-efficient settings of the parameters. On the other hand, it is unclear what can be said about the tightness of the security reductions obtained by Juels and Weis [20] since they do not explicitly handle multiple iterations of the protocol nor do they consider the effect that the acceptance criteria (i.e., in terms of l, u) have on the soundness.

We believe that the security reduction can be improved by taking into account the distribution on $\bar{Z}$ when D's oracle is $A_{\mathbf{s},\varepsilon}$ (and modifying step 3 of D appropriately), as well as by focusing on protocols with perfect completeness. See Section 4 for some discussion of the latter possibility.

3.2 Security of the HB$^+$ Protocol against Active Attacks

The HB protocol is insecure against an active attack, as an adversary can simply repeatedly query the tag with the same challenge vector $(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ and thereby determine with high probability the correct values of $\langle \mathbf{s}, \mathbf{a}_1 \rangle, \ldots, \langle \mathbf{s}, \mathbf{a}_n \rangle$ (after which solving for $\mathbf{s}$ is easy). To combat such an attack, Juels and Weis propose to modify the HB protocol by having the tag and reader share *two* (independent) keys $\mathbf{s}_1, \mathbf{s}_2 \in \{0,1\}^k$. A basic authentication step now consists of three rounds: first the tag sends a random "blinding factor" $\mathbf{b} \in \{0,1\}^k$; the reader replies with a random challenge $\mathbf{a} \in \{0,1\}^k$ as before; and finally the tag replies with $z = \langle \mathbf{s}_1, \mathbf{b} \rangle \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle \oplus \nu$ for $\nu \sim \mathsf{Ber}_\varepsilon$. As in the HB protocol, the tag reader can then verify whether the response z of the tag satisfies $z \stackrel{?}{=} \langle \mathbf{s}_1, \mathbf{b} \rangle \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle$, and we again say the iteration is *successful* if this is the case. See Figure 2.

$$
\begin{array}{rcl}
\underline{\mathcal{T}(\mathbf{s}_1, \mathbf{s}_2, \varepsilon)} & & \underline{\mathcal{R}(\mathbf{s}_1, \mathbf{s}_2)} \\
\mathbf{b} \leftarrow \{0,1\}^k & \xrightarrow{\quad \mathbf{b} \quad} & \\
 & \xleftarrow{\quad \mathbf{a} \quad} & \mathbf{a} \leftarrow \{0,1\}^k \\
\nu \leftarrow \mathsf{Ber}_\varepsilon & & \\
z := \langle \mathbf{s}_1, \mathbf{b} \rangle \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle \oplus \nu & \xrightarrow{\quad z \quad} & \\
 & & \text{verify: } z \stackrel{?}{=} \langle \mathbf{s}_1, \mathbf{b} \rangle \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle
\end{array}
$$

Fig. 2. The basic authentication step of the HB$^+$ protocol.

The actual HB$^+$ protocol consists of n parallel iterations of the basic authentication step (and so the entire protocol requires only three rounds). The protocol also depends upon parameters l, u as in the case of the HB protocol, and the values ε_c and $\delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}$ are defined exactly as there.

The following result characterizes security of the HB$^+$ protocol under *active* attacks. It can be compared to [20, Lemma 3], where Juels and Weis prove security for a single iteration of the HB$^+$ protocol (i.e., they fix $n = 1$). Their proof requires rewinding of the adversary $\mathcal{A}$ in order to simulate the first phase of $\mathcal{A}$, and therefore their proof does *not* extend to the case of parallel or concurrent executions of the basic authentication step.

We remark that by combining the proofs of Theorem 2 and Lemma 1 (i.e., reducing the HB$^+$ protocol directly to the LPN problem rather than relying on Lemma 1 as an intermediate step) we can improve the security reduction stated in the following theorem. By applying techniques from [25, Sect. 4], the parameters of the reduction can be improved further.

Theorem 2. *Say there exists an adversary $\mathcal{A}$ interacting with the tag in at most q executions of the HB$^+$ protocol (possibly concurrently), running in time t, and achieving $\mathsf{Adv}^{active}_{\mathcal{A},\mathsf{HB}^+}(\varepsilon, \mathsf{l}, \mathsf{u}, n) \geq \delta$. Then there exists an algorithm D making $q \cdot n$ oracle queries, running in time $O(t)$, and such that*

$$\left|\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : D^{A_{\mathbf{s},\varepsilon}}(1^k) = 1\right] - \Pr\left[D^{U_{k+1}}(1^k) = 1\right]\right| \\ \geq \left(\frac{\delta + \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}}{2}\right)^3 - \frac{2^n}{2^k} - 2^{-n} \cdot \sum_{i=0}^{2\,\mathsf{u}} \binom{n}{i}.$$

Asymptotically, for any $\varepsilon < \frac{1}{4}$ and appropriate choice of $n, \mathsf{l}, \mathsf{u}$ the last two terms of the above expression (and also ε_c) are negligible. We thus conclude that the HB$^+$ protocol is secure (for appropriate choice of $n, \mathsf{l}, \mathsf{u}$) assuming the hardness of the LPN_ε problem.

Proof. Algorithm D, given access to an oracle returning $(k+1)$-bit strings $(\mathbf{b}, \bar{z})$, proceeds as follows:

1. D chooses $\mathbf{s}_2 \in \{0,1\}^k$ uniformly at random. Then, it runs the first phase of $\mathcal{A}$. To simulate a basic authentication step, D does the following: it obtains a sample $(\mathbf{b}, \bar{z})$ from its oracle and sends $\mathbf{b}$ as the initial message. $\mathcal{A}$ replies with a challenge $\mathbf{a}$, and then D responds with $z = \bar{z} \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle$. Note that since D does not rewind $\mathcal{A}$ here, there is no difficulty in simulating parallel executions of the basic authentication step (i.e., as part of an execution of the "full" HB$^+$ protocol) or concurrent executions of the HB$^+$ protocol.
2. When $\mathcal{A}$ is ready for the second phase of the HB$^+$ protocol, $\mathcal{A}$ sends an initial message $\mathbf{b}_1, \ldots, \mathbf{b}_n$ (we now explicitly look at the actual HB$^+$ protocol rather than focusing on a single basic authentication step). In response, D chooses random $\mathbf{a}^1_1, \ldots, \mathbf{a}^1_n \in \{0,1\}^k$, sends these challenges to $\mathcal{A}$, and records $\mathcal{A}$'s response $z^1_1, \ldots, z^1_n$. Then D rewinds $\mathcal{A}$, chooses random $\mathbf{a}^2_1, \ldots, \mathbf{a}^2_n \in \{0,1\}^k$, sends these to $\mathcal{A}$, and records $\mathcal{A}$'s response $z^2_1, \ldots, z^2_n$.
3. Let $z^\oplus_i := z^1_i \oplus z^2_i$ and set $Z^\oplus \stackrel{\text{def}}{=} \left(z^\oplus_1, \ldots, z^\oplus_n\right)$. Let $\hat{\mathbf{a}}_i = \mathbf{a}^1_i \oplus \mathbf{a}^2_i$ and $\hat{z}_i = \langle \mathbf{s}_2, \hat{\mathbf{a}}_i \rangle$, and set $\hat{Z} \stackrel{\text{def}}{=} (\hat{z}_1, \ldots, \hat{z}_n)$. D outputs 1 iff $Z^\oplus$ and $\hat{Z}$ differ in at most $2\mathsf{u}$ entries.

Let us analyze the behavior of D:

Case 1: Say D's oracle is U_{k+1}. In step 1, above, since $\bar{z}$ is uniformly distributed and independent of everything else, the answers z that D returns to $\mathcal{A}$ are uniformly distributed and independent of everything else. It follows that $\mathcal{A}$'s view throughout the experiment is independent of the secret $\mathbf{s}_2$ chosen by D.

The $\{\hat{\mathbf{a}}_i\}_{i=1}^n$ are uniformly and independently distributed, and so except with probability $\frac{2^n}{2^k}$ they are linearly independent and non-zero (cf. the claim proved below). Assuming this to be the case, $\hat{Z}$ is uniformly distributed over $\{0,1\}^n$ from the point of view of $\mathcal{A}$. But then the probability that $Z^{\oplus}$ and $\hat{Z}$ differ in at most $2\mathsf{u}$ entries is exactly $2^{-n} \cdot \sum_{i=0}^{2\,\mathsf{u}} \binom{n}{i}$. We conclude that D outputs 1 in this case with probability at most $\frac{2^n}{2^k} + 2^{-n} \cdot \sum_{i=0}^{2\,\mathsf{u}} \binom{n}{i}$.

Case 2: Say D's oracle is $A_{\mathbf{s}_1,\varepsilon}$ for randomly-chosen $\mathbf{s}_1$. In this case, D provides a perfect simulation for the first phase of $\mathcal{A}$. By a standard averaging argument, with probability at least $\hat{\delta} \stackrel{\text{def}}{=} \frac{\delta + \delta^*_{\varepsilon,\mathsf{l},\mathsf{u},n}}{2}$ over the randomness used in the first phase of $\mathcal{A}$ (which includes the keys $\mathbf{s}_1, \mathbf{s}_2$, the randomness of $\mathcal{A}$, and the randomness used in responding to $\mathcal{A}$'s queries) the probability (over random challenges $\mathbf{a}_1, \ldots, \mathbf{a}_n$ sent by the tag reader in the second phase) that $\mathcal{A}$ successfully impersonates the tag in the second phase is at least $\hat{\delta}$. Assume this is the case. Then the probability that $\mathcal{A}$ successfully responds to both sets of queries $\mathbf{a}_1^1, \ldots, \mathbf{a}_n^1$ and $\mathbf{a}_1^2, \ldots, \mathbf{a}_n^2$ is at least $\hat{\delta}^2$. But this means that $(z_1^1, \ldots, z_n^1)$ differs in at most u entries from the "correct" answer

$$\mathsf{ans}^1 \stackrel{\text{def}}{=} \left(\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \left\langle \mathbf{s}_2, \mathbf{a}_1^1 \right\rangle, \ldots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \left\langle \mathbf{s}_2, \mathbf{a}_n^1 \right\rangle\right)$$

and also $(z_1^2, \ldots, z_n^2)$ differs in at most u entries from the "correct" answer

$$\mathsf{ans}^2 \stackrel{\text{def}}{=} \left(\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \left\langle \mathbf{s}_2, \mathbf{a}_1^2 \right\rangle, \ldots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \left\langle \mathbf{s}_2, \mathbf{a}_n^2 \right\rangle\right).$$

But then $(z_1^1, \ldots, z_n^1) \oplus (z_1^2, \ldots, z_n^2) = Z^{\oplus}$ differs in at most $2\mathsf{u}$ entries from

$$\begin{aligned}\mathsf{ans}^1 \oplus \mathsf{ans}^2 &= \left(\left\langle \mathbf{s}_2, \mathbf{a}_1^1 \right\rangle \oplus \left\langle \mathbf{s}_2, \mathbf{a}_1^2 \right\rangle, \ldots, \left\langle \mathbf{s}_2, \mathbf{a}_n^1 \right\rangle \oplus \left\langle \mathbf{s}_2, \mathbf{a}_n^2 \right\rangle\right) \\ &= \left(\left\langle \mathbf{s}_2, (\mathbf{a}_1^1 \oplus \mathbf{a}_1^2) \right\rangle, \ldots, \left\langle \mathbf{s}_2, (\mathbf{a}_n^1 \oplus \mathbf{a}_n^2) \right\rangle\right) \quad = \quad \hat{Z}.\end{aligned}$$

We conclude that D outputs 1 in this case with probability at least $\hat{\delta} \cdot \hat{\delta}^2$. This completes the proof of the theorem. $\square$

The following technical claim, used above, is quite straightforward:

Claim. Assume n vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$ are chosen uniformly at random from $\{0,1\}^k$. The probability that these vectors are not linearly independent is less than $\frac{2^n}{2^k}$.

Proof. Say event Bad_i occurs if $\mathbf{a}_i$ is linearly dependent on the previous $i-1$ vectors chosen (for the case $i = 1$ this is the event $\mathbf{a}_1 = 0^k$). Since the subspace spanned by $i-1$ vectors has size at most 2^{i-1}, the probability of Bad_i is at most $\frac{2^{i-1}}{2^k}$. Applying a union bound, we have:

$$\Pr\left[\bigvee_{i=1}^{n} \mathsf{Bad}_i\right] \leq 2^{-k} \cdot \sum_{i=0}^{n-1} 2^i < \frac{2^n}{2^k},$$

yielding the claim. $\square$

A typical range of parameters might be $k \approx 200$ and $n \approx 40$–50, so the $\frac{2^n}{2^k}$ term above is truly inconsequential.

4 Conclusions and Open Questions

The main technical results of this paper are the first rigorous proofs of (1) security of the HB^+ protocol against active attacks, even under parallel and concurrent executions; and (2) "hardness amplification" for the HB and HB^+ protocols as the number of iterations of the basic authentication step increases. Our proofs are also the first to explicitly take into account the non-zero completeness error and the impact this has on the security of the protocol as a whole.

We believe our proofs are remarkably simple, and view this as an additional contribution of this work (rather than as a drawback!). Indeed, we expect there will be further applications of Lemma 1 to the analysis of other cryptographic constructions based on the LPN problem, and hope this paper inspires and aids others in exploring such applications.

It would be nice to improve the analysis (or propose new protocols) so as to obtain meaningful security guarantees even in the case $\frac{1}{4} \leq \varepsilon < \frac{1}{2}$. It would also be wonderful to improve the concrete security reductions obtained here, or to propose new protocols with tighter security reductions. (As we have mentioned, it is not clear whether the proofs provided here yield sufficiently-high security for practically-efficient settings of the parameters.) As one possible approach toward this goal, one can imagine changing the $\mathrm{HB}/\mathrm{HB}^+$ protocols so that the tag always introduces *at most* $\varepsilon \cdot n$ errors, rather than introducing errors in each of the n iterations with independent probability ε.[7] (A related idea, in a different context, was explored in [5]; their analysis does not seem to apply to our setting.) This would give protocols with perfect completeness, and would help improve the concrete security bounds as well since the upper bound u could be set to exactly $\varepsilon \cdot n$ and the "problem" mentioned in footnote 5 would also go away. On the other hand it is not clear what can be said of the hardness of the natural variant of the LPN problem such protocols would be based on.

It would also be very interesting to see an efficient protocol based on the LPN problem that is provably resistant to man-in-the-middle attacks.

References

1. Associated Press. "Geeks Flex Hacker Muscles at Defcon." Article appeared Aug. 2, 2005 on CNN.com.
2. M. Bellare, M. Fischlin, S. Goldwasser, and S. Micali. Identification Protocols Secure against Reset Attacks. *Adv. in Cryptology — Eurocrypt 2001*, LNCS vol. 2045, Springer-Verlag, pp. 495–511, 2001.
3. M. Bellare, R. Impagliazzo, and M. Naor. Does Parallel Repetition Lower the Error in Computationally-Sound Protocols? *38th IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 374–383, 1997.
4. E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg. On the Inherent Intractability of Certain Coding Problems. *IEEE Trans. Info. Theory* 24: 384–386, 1978.

[7] Note that introducing *exactly* $\varepsilon \cdot n$ errors in the n iterations is insecure.

5. A. Blum, M. Furst, M. Kearns, and R. Lipton. Cryptographic Primitives Based on Hard Learning Problems. *Adv. in Cryptology — Crypto '93*, LNCS vol. 773, Springer-Verlag, pp. 278–291, 1994.
6. A. Blum, A. Kalai, and H. Wasserman. Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. *J. ACM* 50(4): 506–519, 2003.
7. R. Canetti, S. Halevi, and M. Steiner. Hardness Amplification of Weakly Verifiable Puzzles. *2nd Theory of Cryptography Conference (TCC 2005)*, LNCS vol. 3378, Springer-Verlag, pp. 17–33, 2005.
8. R. Canetti, J. Kilian, E. Petrank, and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires (Almost) Logarithmically Many Rounds. *SIAM J. Computing* 32(1): 1–47, 2002.
9. F. Chabaud. On the Security of Some Cryptosystems Based on Error-Correcting Codes. *Adv. in Cryptology — Eurocrypt '94*, LNCS vol. 950, Springer-Verlag, pp. 131–139, 1995.
10. W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Trans. Info. Theory* 22(6): 644–654 (1976).
11. U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. *22nd ACM Symposium on Theory of Computing*, ACM, pp. 416–426, 1990.
12. H. Gilbert, M. Robshaw, and H. Silbert. An Active Attack against HB^+ — a Provably Secure Lightweight Authentication Protocol. Available at http://eprint.iacr.org/2005/237
13. O. Goldreich. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness.* Springer-Verlag, 1998.
14. O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM J. Computing* 25(1): 169–192, 1996.
15. O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR-Lemma. Available at http://eccc.uni-trier.de/eccc-reports/1995/TR95-050/
16. O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *J. Cryptology* 7(1): 1–32, 1994.
17. J. Håstad. Some Optimal Inapproximability Results. *J. ACM* 48(4): 798–859, 2001.
18. N. Hopper and M. Blum. A Secure Human-Computer Authentication Scheme. Technical Report CMU-CS-00-139, Carnegie Mellon University, 2000.
19. N. Hopper and M. Blum. Secure Human Identification Protocols. *Adv. in Cryptology — Asiacrypt 2001*, LNCS vol. 2248, pp. 52–66, 2001.
20. A. Juels and S. Weis. Authenticating Pervasive Devices with Human Protocols. *Adv. in Cryptology — Crypto 2005*, LNCS vol. 3621, Springer-Verlag, pp. 293–308, 2005. Updated version available at: http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/publications/pdfs/lpn.pdf
21. M. Kearns. Efficient Noise-Tolerant Learning from Statistical Queries. *J. ACM* 45(6): 983–1006, 1998.
22. Z. Kfir and A. Wool. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems. Available at http://eprint.iacr.org/2005/052
23. I. Kirschenbaum and A. Wool. How to Build a Low-Cost, Extended-Range RFID Skimmer. Available at http://eprint.iacr.org/2006/054
24. R. Raz. A Parallel Repetition Theorem. *SIAM J. Computing* 27(3): 763–803, 1998.
25. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *37th ACM Symposium on Theory of Computing*, ACM, pp. 84–93, 2005.
26. A. C.-C. Yao. Theory and Applications of Trapdoor Functions. *23rd IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 80–91, 1982.